Reactive Rearrangement of Parts under Sensor Inaccuracy: Particle Filter Approach

Halûk Bayram*, Ayşın Ertüzün† and H. Işıl Bozma†
Intelligent Systems Laboratory
*System and Control Engineering, †Electrical and Electronic Engineering
Bogazici University, Bebek 34342 Istanbul Turkey

***Abstract*— The paper addresses the warehouseman's problem with geometrical simplifications under the more realistic case of imperfect sensory information. In this scenario, a 2D workspace contains an actuated robot and a set of unactuated parts. The discrepancy between the robot's and/or the parts' real and measured positions may lead to jerky movements or even collisions in the parts' moving problem we are concerned with. Thus, we need to approximate the state information – taking the highly nonlinear nature of the resulting system into account. This is accomplished using particle filters – which implement recursive Bayesian filter in nonlinear and/or nongaussian environments. For the model of parts which turns out to be linear, the approach reduces to Kalman filtering. First the robot's dynamic model and the measurement model are modified to incorporate the inaccuracies in the sensory data; and then the particle filter is utilized to get improved positional estimate. Enhancements in the robot's movements and reduction in the number of collisions have been verified through extensive computer simulations.**

I. Introduction

This paper addresses the warehouseman's problem [1] - still with geometrical simplifications [2] but no longer assuming perfect sensory information. In this scenario, a 2D workspace contains a disk-shaped robot and a set of unactuated parts - all disk-shaped. The robot's task is to move all the parts to their prespecified goal positions. As there is no guarantee that the parts will be left undisturbed, the robot is required to employ a reactive strategy. A feedback-based event-driven approach based on artificial potential functions has been presented in [2]. However, their formal algorithm is predicated on the assumptions that the robot (i) has perfect (online) knowledge of the locations of the parts, and (ii) has perfect (online) knowledge of its joint positions. As well known, in actual implementations, these assumptions may not hold as sensory measurements are fed back from the optical encoders and camera-based vision system which are both prone to sensor inaccuracies. The discrepancy between the robots' real and measured positions may lead to jerky movements and even possibly collision. Hence, the robot cannot operate purely based on raw sensory data as has been the case, but now needs to estimate positional information. The contribution of this paper is to consider the more realistic case of sensory inaccuracy and integrate particle filters (PF) with reactive strategies based on artificial potential functions.

A. Related Literature

One approach to achieving reactivity has been based on artificial potential functions [8], [9]. However, their usage has been limited since most constructions suffer from undesired local minima and thus are not ensured of convergence to the desired goal positions. A sequel of work reported has then been able to demonstrate that it is possible to overcome this shortcoming through carefully constructed functions. Navigation functions have been shown to have global convergence properties and a construction with this property for the case of robot navigation among static obstacles has been presented in [11]. The approach has then been generalized to linear parts' moving or rearrangment problems where it has been shown that by sequentially switching among a family of feedback controllers, we can generate a plan that is ensured of convergence or termination [10]. An actual implementation in 2D, as presented in [2] has demonstrated the robustness of the approach in case of dynamically changing environments. However, in all this work, sensory information has been perfect.

Robustness against sensor noise is an important issue in most robotic applications. PFs have been proposed as an probabilistic approach for estimating the state of the dynamic system – for the case of highly nonlinear problems in [4], [6]. Several variants of the PF such as sampling importance resampling (SIR) filter, auxiliary sampling importance resampling (ASIR) filter, and Rao-Blackwellized PF have then been introduced within a generic framework of the sequential importance sampling (SIS) algorithms [14]. Their application in robotic problems has been studied by many researchers [3], [12], [13] where Bayesian filtering with particle-based density representations are used to localize a mobile robot. Their advantage stems from the fact that they solve the associated optimal filtering problem -which in general does not admit a closed-form solution- numerically [5].

B. Problem Statement

An actuated robot inhabits 2D workspace along with a set of unactuated rigid bodies. All the positions including that of itself are a priori unknown. They are imperfectly sensed due to sensor and measurement noise. The robot should react to the sensed arrangement of parts based on this inaccurate data by choosing a particular one to move, then choose a next part after another look at the new arrangement based on updated

sensory readings and so on, in such a fashion as eventually to move them all into their specified destinations.

C. Approach

Our approach is based on integration of PFs with feedback-based event driven approach. First, the robot's dynamic model is modified to incorporate sensor inaccuracies. Next, PFs are constructed in order to get an improved positional estimate of the robot and the other parts. The robot's position is updated using the pdf thus constructed. As the parts are stationary by themselves, PFs simplify out to be Kalman filters which are then used to improve the estimate of the parts' positions.

II. Feedback-Based Event-Driven Parts Moving

The mathematical formalism of feedback-based event-driven parts' moving is as follows [2]: Each part $i \in P = \{1, \cdots, p\}$, $p \in Z^+$, is defined by its center $b_i \in R^2$, and radius $\rho_i \in R$. The state vector of all of the parts $b \in R^{2p}$ is defined as $b = \sum_{i \in P} b_i \otimes e_i$, where $e_i \in R^p$ are the unit vectors in R^p. The goal of each player i is to be moved to its goal position $g_i \in R^2$ without colliding with other players. The vector of all the goal positions $g \in R^{2p}$ is defined as $g = \sum_{i \in P} g_i \otimes e_i$. The robot is defined by its center $r \in R^2$, angle of its gripper with respect to the x axis θ and radius $\rho_r \in Z^+$. The augmented robot state vector is defined as $r_a = SE(2)$ as $r_a = [r\ \theta]^T$. The problem is solved with the following rules:

- For each part, a subtask is assigned as a sequence of two stages: i) *mate-part*: Robot moves to mate with part, ii) *move-part*: Robot moves the part to its goal position.
- One subtask gets to be executed at a time.
- Subtasks are competing and the robot selects the subtask based on an urgency measure.

A. Robot Part Mating

The mating control laws are defined by a collection of smooth scalar valued maps $\varphi_i : R^p \times R^{2p} \rightarrow R$, $\forall i \in P$. Each $\varphi_i(r, b)$ is defined as:

$$\varphi_i(r, b) = \frac{\gamma_i^{k_2}(r, b)}{\beta_r(r, b)}$$

where $\gamma_i : R^2 \times R^{2p} \rightarrow R$ is the squared Euclidian distance between the robot and part i defined as $\gamma_i(r, b) = \|r - b_i\|^2$, $\forall i \in P$. The obstacle function $\beta_i(r, b) : R^2 \times R^{2p} \rightarrow R$ is defined as $\beta_i(r, b) = \prod_{i \in P} \|r - b_i\|^2 - (\rho_i - \rho_j)^2$, $\forall i \in P$. The constant $k_2 \in Z^+$ is a positive integer chosen appropriately. The robot's motion toward part i is governed by the dynamical system:

$$\dot{r} = -D_r \varphi_i(r, b)$$

B. Robot Part Moving

Once the robot mates with part i, the robot-part coupled structure moves as a single body in the extended space SE(2). The position vector b_i of part i is dependent on the augmented state vector r_a as follows: $b_i = r + d \begin{bmatrix} cos\theta \\ sin\theta \end{bmatrix}$ where d is the mating distance between the robot and the mated part.

Control laws are defined by again considering a collection of smooth maps of $\psi_i : SE(2) \times R^{2p-2} \rightarrow R$, $\forall i \in P$ defined as:

$$\psi_i(r_a, \overline{b}_i) = \frac{\gamma_i^{k_3}(r_a, \overline{b}_i)}{\beta_i(r_a, \overline{b}_i)}$$

where $\overline{b}_i = \{b_1, ..., b_{i-1}, b_{i+1}, ..., b_p\}$. The parameter $k_3 \in Z^+$ is chosen to be a positive integer. The function $\gamma(r_a, \overline{b}_i) : SE(2) \times R^{2p-2} \rightarrow R$ is the total squared Euclidean distance between the current configuration and the goal configuration:

$$\gamma(r_a, \overline{b}_i) = \left\| r + d \begin{bmatrix} cos\theta \\ sin\theta \end{bmatrix} - g_i \right\|^2 + \sum_{i \in P}^{j \neq i} \|b_j - g_j\|^2$$

The function $\beta_i(r_a, \overline{b}_i) : SE(2) \times R^{2p-2} \rightarrow R$, $\forall i \in P$, encodes the obstacles as:

$$\beta_i(r_a, \overline{b}_i) = \prod_{j \in P}^{j \neq i} \left[\|r - b_j\|^2 - (\rho_r - \rho_j)^2 \right]$$

$$\times \left[\left\| r + d \begin{bmatrix} cos\theta \\ sin\theta \end{bmatrix} - b_j \right\|^2 - (\rho_i - \rho_j)^2 \right]$$

The motion of robot carrying part i is defined by:

$$\dot{r}_a = -D_{r_a} \psi_i(r_a(t), \bar{b}_i(t)) \tag{1}$$

C. Next Part

The robot chooses among the competing subtasks using a index valued *next-part* function $h(b) : R^{2p} \rightarrow P$:

$$h(b) = \arg\max_{i \in P} \left\| \left(I_2 \otimes e_i^T \right) D_b \phi(b) \right\|$$

This function picks out the component of b - whose direction of descent on $\phi(b)$ is the steepest. The function $\phi(b) : R^{2p} \rightarrow R$ is defined as $\phi(b) = \left(\gamma^{k_1}(b) / \beta(b) \right)$ where $k_1 \in Z^+$. The term $\gamma(b) : R^{2p} \rightarrow R$ is defined as $\gamma(b) = \|b - g\|^2$ while the denominator $\beta(b) : R^{2p} \rightarrow R$ denotes the obstacle function of the pairwise part positions defined as $\beta(b) = \prod_{i \in P} \prod_{j \in P}^{j > i} \|b_i - b_j\|^2 - (\rho_i - \rho_j)^2$.

III. State and Observation Models

The aim is to take noise in the sensory readings into account based on particle and Kalman filters and to enable the robot with improved navigation under noisy sensing.

A. Robot Motion State and Observation Model

Consider the robot state $r(t)$ at time t. Let η_d and η_m represent dynamical and measurement noise respectively. Assume that both are Gaussian $\eta_d \sim N(0, \Sigma_d)$, $\eta_m \sim N(0, \Sigma_m)$, where the covariances Σ_d and Σ_m are also known. Under noisy measurements, the state dynamics for the *mate-part* now incorporate dynamical noise as:

$$\dot{r}(t) = -D_r \varphi_i(r(t), b(t)) + \eta_d$$

In the *move-part* stage, similarly the system dynamics should now incorporate noise as well:

$$\dot{r}(t) = -D_{r_a}\psi_i(r_a(t), \bar{b}_i(t)) + \eta_d$$

Due to measurement noise, the robot state $r(t)$ is no longer available. Rather, we have its noisy measurement $z(t)$.

$$z(t) = r(t) + \eta_m$$

B. Part State and Observation Model

Parts' state model is linear because parts are stationary unless the robot carries them. The system model is as follows:

$$\dot{\bar{b}}_i(t) = \eta_d$$

Similarly, we have only observations $z_{\bar{b}_i}$ of $\bar{b}_i$

$$z_{\bar{b}_i}(t) = \bar{b}_i(t) + \eta_m$$

IV. Uncertainty Reduction By Particle Filter and Kalman Filter

A. Particle Filter

Particle filter is a Bayesian filter for implementing Markov Chain Monte Carlo Methods (MCMC) sequentially. It approximates the a posteriori pdf, under nonlinear and nongaussian system dynamics, based on a finite number of samples with associated weights [4]. It provides an estimate $\hat{r}_t$ of r_t – which is discretized $r(t)$ at time $t = k\Delta t$. This is accomplished by weighted approximation of the posterior density function (pdf) $p(r_t|z_t)$ from samples z_t of $z(t)$ in a recursive manner. There are two stages: prediction and updating. In prediction, the $\hat{r}_t$ is predicted from the previous observations. Following, the posterior pdf $p(r_t|z_t)$ is updated using the newly acquired measurements. It is represented using a set consisting of N particles $\left\{r_t^i \mid i = 1, \cdots, N\right\}$ with associated weights w_t^i, for each time step t as:

$$p(r_t|z_t) \approx \sum_{i=1}^{N} w_t^i \delta(r_t - r_t^i) \tag{2}$$

It has been shown that as $N \rightarrow \infty$, the approximation approaches the true posterior [4]. The weights are updated within a normalization using the importance sampling as :

$$w_t^i \propto w_{t-1}^i \frac{p(z_t|r_t^i)p(r_t^i|r_{t-1}^i)}{q(r_t^i|r_{0:t-1}^i, z_{0:t})} \tag{3}$$

where $r_{0:t-1} = [r_0, r_1, .., r_{t-1}]^T$ and $z_{0:t} = [z_0, z_1, .., z_t]^T$. Note that the state equation characterizes the state transition probability $p(r_t|r_{t-1})$, whereas the measurement equation describes the likelihood $p(z_t|r_t)$. The pdf $q(r_t^i|r_{0:t-1}^i, z_{0:t})$ is referred to as the importance or proposal density (function). Its choice is critical with respect to performance. The optimal importance function that minimizes the variance of the importance weights, conditioned on $r_{0:t-1}$ and $z_{0:t}$ has been shown to be [6]:

$$\begin{aligned} q(r_t|r_{0:t-1}^i, z_{0:t})_{opt} &= p(r_t|r_{t-1}^i, z_t) \\ &= \frac{p(z_t|r_t, r_{t-1}^i)p(r_t|r_{t-1}^i)}{p(z_t|r_{t-1}^i)} \end{aligned} \tag{4}$$

The substitution of (4) into (3) yields

$$w_t^i \propto w_{t-1}^i p(z_t|r_{i-1}^i) \tag{5}$$

It requires sampling from $p(r_t|r_{t-1}^i, z_t)$ and evaluation of $p(z_t|r_{t-1}^i)$ - which are both realizable in our case. Under our assumptions on the state and observation models, both the optimal importance density and $p(z_t|r_{t-1})$ are Gaussian, i.e.:

$$\begin{aligned} p(r_t^i|r_{t-1}^i, z_t) &= N(r_t^i; m_t, \Sigma_t) \\ p(z_t|r_{t-1}^i) &= N(z_t; f_{t-1}, \Sigma_d + \Sigma_m) \end{aligned} \tag{6}$$

where

$$\begin{aligned} \Sigma_t^{-1} &= \Sigma_d^{-1} + \Sigma_m^{-1} \\ m_t &= \Sigma_t \left(\Sigma_d^{-1} f_{t-1} + \Sigma_m^{-1} z_t\right) \end{aligned}$$

$$f_t = \begin{cases} -D_r\varphi_i(r_t, b_t) & \text{if } state = mate - part \\ -D_{r_a}\psi_i(r_{a_t}, \bar{b}_{i_t}) & \text{if } state = move - part \end{cases} \tag{7}$$

b_t is discretized $b(t)$ at time $t = k\Delta t$. The weights at time t can be computed and normalized such that $\sum_i w_t^i = 1$ before the particles are propagated to time t. One can perform resampling ,if necessary, to obtain an approximate i.i.d. sample from $p(r_t|z_t)$.

This algorithm forms the basis for the PF which consists of the recursive evolution of the importance weights and the particles as measurements are received sequentially. A pseudo-code description of the algorithm is given in Table I.

Without resamling in step-5 of Table-I, during the computation of the weights, most particles may converge to 0 which is a common problem known as the degeneracy phenomenon. Hence, the filter would not estimate $\hat{r}_t$ from noisy observations properly. The particles are resampled whenever a significant degeneracy is observed ,that is, whenever N_{eff} falls below a certain threshold N_{th}. The threshold was chosen as $N_{th} = 2N/3$. The effective sample size N_{eff} introduced in [7] is estimated as:

$$N_{eff} = \frac{1}{\sum_{i=1}^{N} \left(w_t^i\right)^2} \tag{8}$$

In the resampling stage, N particles are taken with replacement from $\left\{r_t^i\right\}_{i=1}^N$, where the probability to take particle i is w_t^i. Systematic resampling [4] is used in step-5 of Table-I.

B. Kalman Filter

The Kalman Filter is used to provide an estimate $\hat{b}_t$ of b_t. It is summarized in Table-II where K is called the gain matrix, and P is called the covariance of the prediction error. Like the PF, there are two stages in the estimation procedure-the prediction and the update. However, the Kalman filter assumes,

TABLE I

SIR PARTICLE FILTER

1) Initialization
   For i = 1:N
   - Draw $r_t^i \sim N(z_0, \Sigma_m)$

   End For
2) Sample from the optimal importance density
   For i = 1:N
   - Draw $r_t^i \sim N(r_t^i; m_t, \Sigma_t)$
   - Assign the particle a weight, $\tilde{w}_t^i$, according to eq. (5)

   End For
3) Compute normalized importance weights
   For i = 1:N
   - $w_t^i = \tilde{w}_t^i / SUM\left[\left\{\tilde{w}_t^i\right\}_{i=1}^N\right]$

   End For
4) Calculate N_{eff} according to eq. (8)
5) IF $N_{eff} < N_{th}$
   Resample using systematic resampling
   $\left\{r_t^i, w_t^i\right\}_{i=1}^N = RESAMPLE\left[\left\{r_t^i, w_t^i\right\}_{i=1}^N\right]$
6) Estimate $\hat{r}_t$, weighted sum of particles
7) Simulate the system using estimated $\hat{r}_t$
8) Go to step-2

TABLE II

KALMAN FILTER

1) Prior estimate $\hat{b}_t^-$, error covariance P_t^-
2) Compute Kalman gain
   $K_t = P_t^-\left(P_t^- + \Sigma_m\right)^{-1}$
3) Update estimate with measurement z_b
   $\hat{b}_t = \hat{b}_t^- + K_t\left(z_{b,t} - \hat{b}_t^-\right)$
4) Compute error covariance for updated estimate
   $P_t = (I - K_t)\, P_t^-$
5) Project ahead
   $\hat{b}_{t+1}^- = \hat{b}_t$
   $P_{t+1}^- = P_t + \Sigma_d$
6) Go to step-2

at every time step t, that the posterior density is Gaussian and hence characterized by a mean and covariance. It can be proved that if $p(b_{t-1}|z_{t-1})$ is Gaussian then $p(b_t|z_t)$ is also Gaussian provided that certain assumptions hold: The process and the measurement noise are both Gaussian distributions of known parameters and both the state equation and the measurement equation are known linear functions of the relevant states and noise [15]. These assumptions are all satisfied for the parts' state and observation models in our case. With the assumption of a prior estimate $\hat{b}_t^-$ and a prior error covariance P^-, the measurement z_b is used to improve the prior estimate as in step-3 of Table-II. The updated estimated $\hat{b}_t$ is easily projected ahead via the transition matrix which is a identity matrix in our case given as in step-5 of Table-II.

The Kalman filter algorithm can be summarized as follows [4]:

$$\begin{aligned} p(b_{t-1}|z_{b,1:t-1}) &= N(b_{t-1}; \hat{b}_t, P_{t-1}) \\ p(b_t|z_{b,1:t-1}) &= N(b_t; \hat{b}_t^-, P_t^-) \\ p(b_t|z_{b,1:t}) &= N(b_t; \hat{b}_{t+1}, P_t) \end{aligned}$$

V. SIMULATIONS

A series of simulations have been conducted using 3 and 6 parts. Task complexity is varied from easy (E) to intermediate (I) and to finally hard (H) where the classification is based on the difficulty of the task as measured by how tightly packed the parts need to be at their final positions. The measurement noise level is selected as low noise (variance = 0.2), medium noise (variance = 1.0) and high noise (variance = 5.0) according to SNR (Signal to Noise Ratio). The process noise variance is 0.06 . In order to evaluate performance where the unit is in cm, first the simulations are performed without any state estimation under different noise levels. Each experiment is repeated with 10 random initial configurations. Next, same experiments are repeated with the number of particles varying as 49, 100 and 225. Similar to measures as presented in [2], the performance measures are as follows: normalized robot length (nrl), normalized part length (npl), positional inaccuracy (pi), robot positional estimation error (rpee), parts positional estimation error (ppee) and the collision percentage. Their definitions are:

Normalized part length (npl):

$$npl = \frac{1}{p}\sum_{i\in P}\frac{\int_0^{t_f}\left\|\dot{b}_i(t)\right\|dt}{\|b_i(0) - g_i\|}$$

where t_f denotes the duration of a task.

Normalized robot part length (nrl):

$$nrl = \frac{\int_0^{t_f}\|\dot{r}_i(t)\|\,dt}{\sum_{i\in P}\|r(0) - b_i(0)\| - (\rho_i + \rho_r) + \sum_{(i,j)\in P}\|g_i - g_j\|}$$

where $r(0)$ denotes the initial position of the robot.

Positional inaccuracy (pi):

$$pi = \frac{1}{p}\sum_{i\in P}\frac{1}{\rho_i}\|b_i(t_f) - g_i\|$$

Robot positional estimation error (rpee):

$$rpee = \frac{1}{t_f}\sum_{t=0}^{t_f}\|r(t) - \hat{r}_t\|$$

Part positional estimation error (ppee):

$$ppee = \frac{1}{t_f}\sum_{t=0}^{t_f}\left\|b(t) - \hat{b}_t\right\|$$

The nrl, npl and pi values are further normalized with respect to their associated values at no noise conditions. Hence, their closeness to 1 indicate similarity of performance to ideal conditions.

In the figures, each data point represents the mean of 10 sample run with random initial configuration. The posterior pdf is supergaussian as associated kurtosis is calculated at each iteration and found to be in the range of 1.5 to 2. This verifies that conventional Kalman filtering cannot be utilized in our highly nonlinear system.

The simulation results for 3-part and 6-part arrangement problems are as shown in Fig.1-3 and Fig.4-6. For low noise,

TABLE III

COLLISION PERCENTAGES IN 3-PART

Num. of Particle	Low Noise			Medium Noise			High Noise		
	E	I	H	E	I	H	E	I	H
0 (No PF)	0	10	10	0	30	30	10	40	50
49	0	0	0	0	10	10	0	10	20
100	0	0	0	0	0	10	0	10	20
225	0	0	0	0	0	0	0	0	10

it is observed that there is only slight improvement in performance by using PF. Performance does not vary much with added task complexity. Furthermore, increasing the number of particles does not change the performance – in particular with respect to estimation errors in the robot's and parts' positions as the noise is not high enough to deteriorate the performance. In medium and high noise, a notable decrease in variance of rpee occurs, which results in lowered nrl, npl and pi values - particular for the case of 225 particles. This is expected since there is a considerable improvement in the estimate of the instantaneous state information with the particle and Kalman filters. The positional error in the parts' position is nearly the same in all task difficulty levels and is independent from the number of particles because Kalman filtering is applied to estimate parts' position.

The collision percentages for each task are shown in Table-III and Table-IV respectively for the case of 3 and 6 parts. As expected, the collision percentages increase as a function of task difficulty and noise level. Furthermore, increased number of particles decrease the number of collisions considerably.

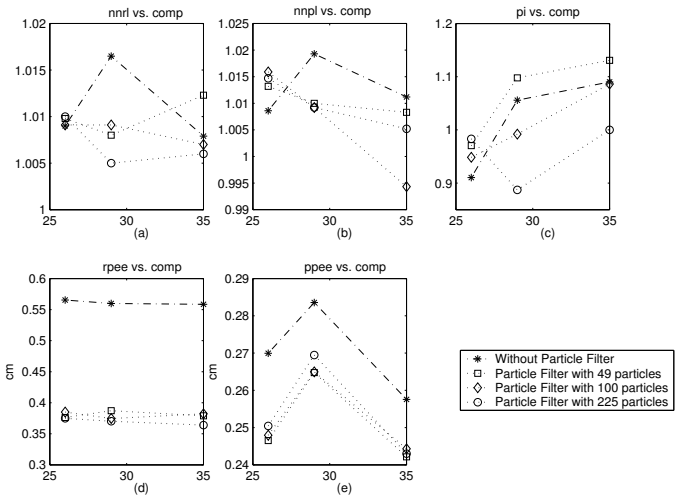

Fig. 1. Performance measures vs. task complexity in 3-Part , Low Noise

VI. CONCLUSION

This paper studies the feedback-based version of the parts rearrangement problem under sensor inaccuracy. In this scenario, a 2D robot inhabits the same workspace as a set of parts which need to be moved from an arbitrary initial placement to a final goal configuration. As the parts' positions

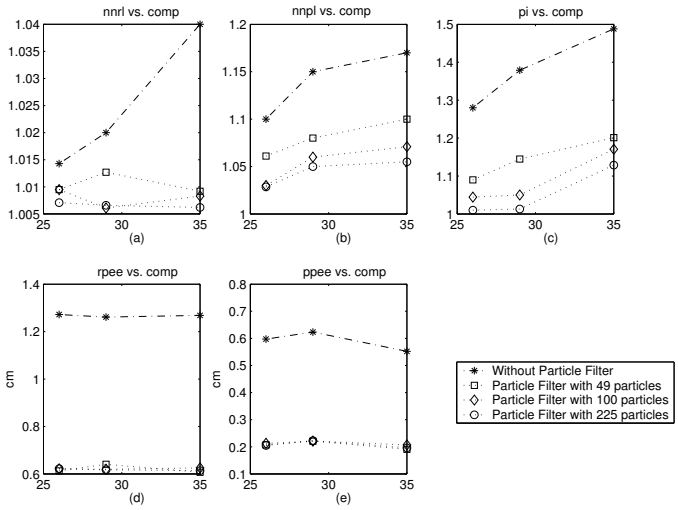

Fig. 2. Performance measures vs. task complexity in 3-Part , Medium Noise

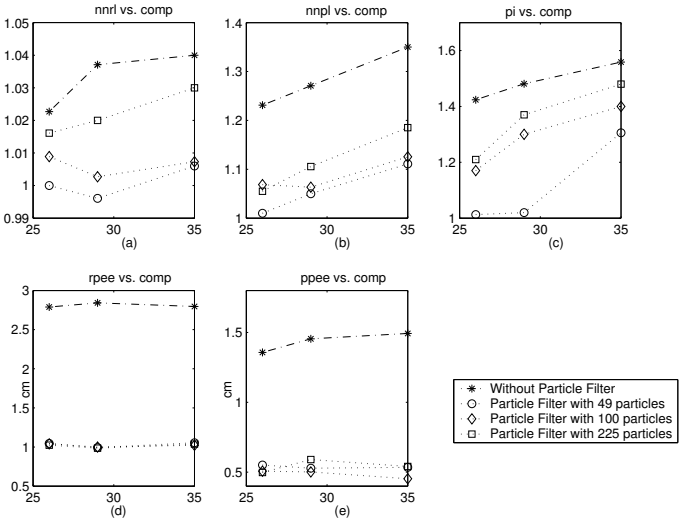

Fig. 3. Performance measures vs. task complexity in 3-Part, High Noise

may be disturbed along the way, we employ a feedback-based approach based on artificial potential functions and non-cooperative games. However, unlike previous work, we no longer assume perfect sensory knowledge as will be the case in many applications. The models used for the dynamics and measurement are modified to include noise. As the resulting system has both nonlinear and linear parts, the states are estimated using particle and Kalman filters. The capability to handle nonlinear, non-Gaussian systems allows PFs to achieve improved accuracy over Kalman filter-based estimation methods. The more nonlinear the model is, or the more non-Gaussian the noise is, the more potential the PF has. The simulation results indicate that in cases of increased sensor inaccuracy, the robot's movements and the positional inaccuracy are both improved. More importantly, the number of collisions and the percentage of successful task completions are improved dramatically.

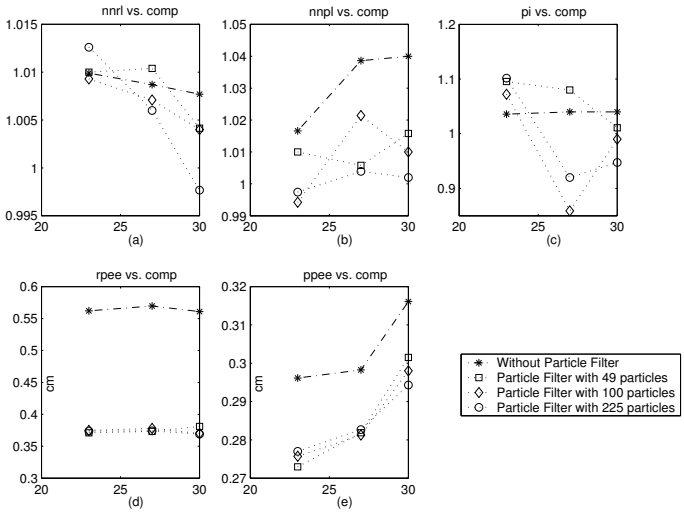

Fig. 4. Performance measures vs. task complexity in 6-Part , Low Noise

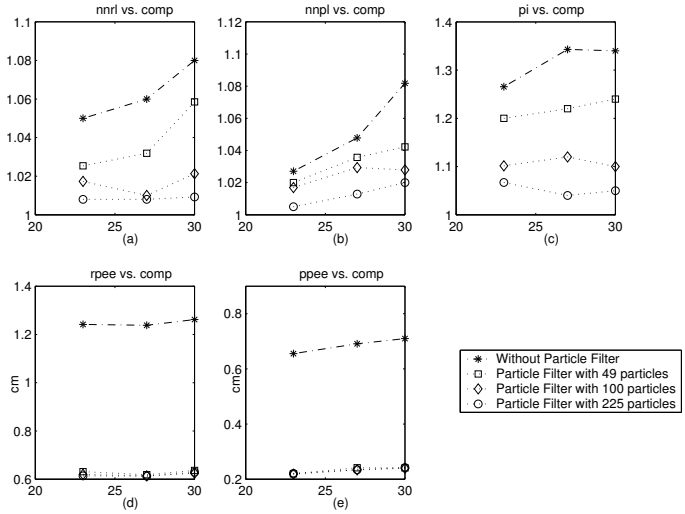

Fig. 5. Performance measures vs. task complexity in 6-Part , Medium Noise

ACKNOWLEDGMENT

This work has been supported by Bogazici University BAP Project 05A202 and DPT Project 03K120250. The first author is supported through a grant from TUBITAK-BAYG program.

REFERENCES

[1] Hopcroft,J.E., J.T. Schwartz, and M. Sharir, *On the complexity of motion planning for multiple independent objects: PSPACE-hardness of the warehouseman's problem*, Int. J. Robot. Res., vol. 3, no. 4, pp. 76-88, 1984.

[2] Karagöz, C.S., H.I. Bozma, and D.E. Koditschek, *Feedback-based event-driven parts moving*, IEEE Trans. Robotics, Vol 20, No. 6, pp:1012-1018, 2004.

[3] Dellaert, F., D. Fox, W. Burgard, and S. Thurn, *Monte Carlo localization for mobile robots*, Proc. IEEE Int. Conf. on Robotics and Automation (ICRA), 1999.

[4] Arulampalam, M.S., S. Maskell, N. Gordon, and T. Clapp, *A tutorial on particle filters for online non-linear/non-Gaussian Bayesian tracking*, IEEE Trans. Signal Processing, vol. 50, no. 2, pp. 174-188, 2002.

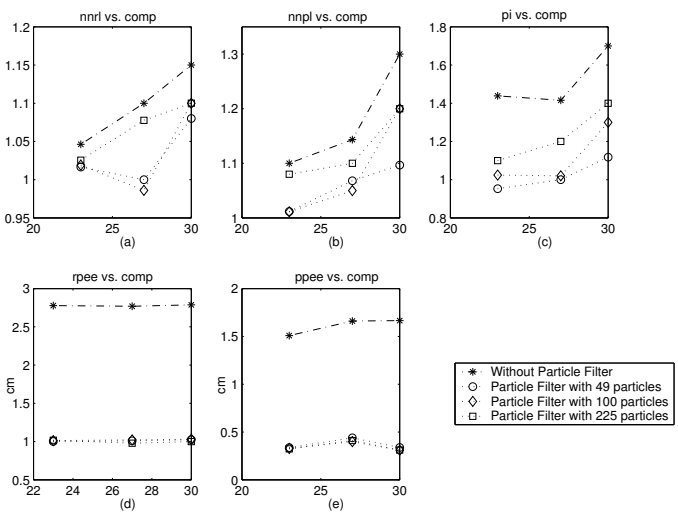

Fig. 6. Performance measures vs. task complexity in 6-Part, High Noise

TABLE IV
COLLISION PERCENTAGES IN 6-PART

Num. of Particle	Low Noise			Medium Noise			High Noise		
	E	I	H	E	I	H	E	I	H
0 (No PF)	0	30	50	0	50	60	10	70	70
49	0	10	30	0	10	30	0	20	40
100	0	10	20	0	10	20	0	20	30
225	0	0	10	0	0	10	0	10	20

[5] Crisan, D., A. Doucet, *A survey of convergence results on particle filtering for practitioners*, IEEE Trans. Signal Processing, vol. 50, no. 3, pp. 736-746, 2002.

[6] Doucet, A., S. Godsill, and C. Andrieu, *On sequential Monte Carlo sampling methods for Bayesian filtering*, Statistics and Computing, vol. 10, no. 3, pp. 197-208, 2000.

[7] Liu, J.S. and R. Chen, *Sequential Monte Carlo methods for dynamical systems*, J. Amer. Statist. Assoc., vol. 93, pp. 1032-1044, 1998.

[8] Khatib, O., *Real time obstacle avoidance for manipulators and mobile robots*, Int. J. of Robotics Research, Vol. 5, No. 1, pp. 90-99, 1986.

[9] Koditschek, D.E., *The application of total energy as a Lyapunov function for mechanical control systems*, in Control Theory and Multibody Systems, J. Marsden, P.S. Krishnaprasad, and J. Simo, Editors., American Mathematical Society. p. 131-158, 1989.

[10] Bozma, H. I. and D. E. Koditschek, *Assembly as a noncooperative game of its pieces: analysis of 1D sphere assemblies*, *Robotica*, Vol. 19, pp. 93-108, 2001.

[11] Rimon, E. and Koditschek, D. E., *Exact robot navigation using artificial potential functions*, *IEEE Trans. on Robotics and Automation*, Vol.8, No.5, pp. 501-518, 1992.

[12] Thrun, S. , *Particle filters in robotics*, Proceedings of the 17th Annual Conference on Uncertainty in AI (UAI), pp:511-518 , 2002.

[13] Gustafsson, F., F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson and P. Nordlund, *Particle filters for positioning, navigation and tracking*, IEEE Trans. on Signal Processing, Vol. 50, No 2, pp: 425-437, 2002.

[14] Rystic, B., S. Arulampalam, N. Gordon, *Beyond Kalman Filter Particle Filter for Tracking Applications*, Artech House, Boston, 2004.

[15] Haykin, S., *Adaptive Filter Theory* , Prentice-Hall, 4th ed., 2001.